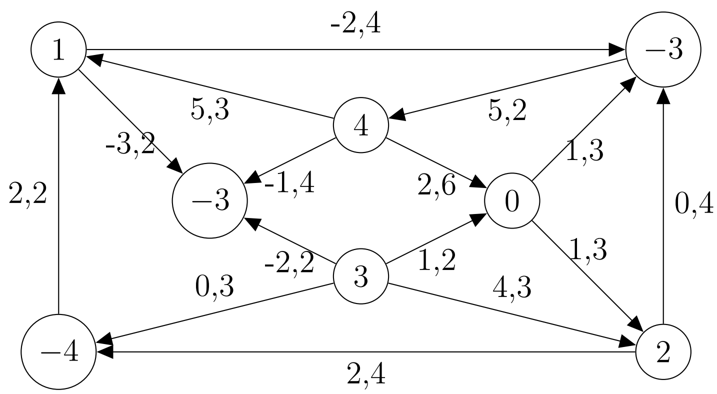
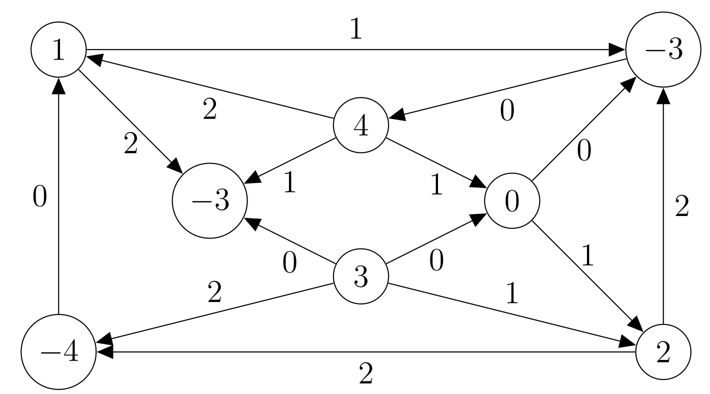
MATH-566 HW 10

Due **Nov 11** before class (regularly). Just bring it before the class and it will be collected there. If you need extra time (say HW from graph theory), bring it by Nov 13.

Consider the following network M with costs and capacities depicted on edges and boundary in vertices.



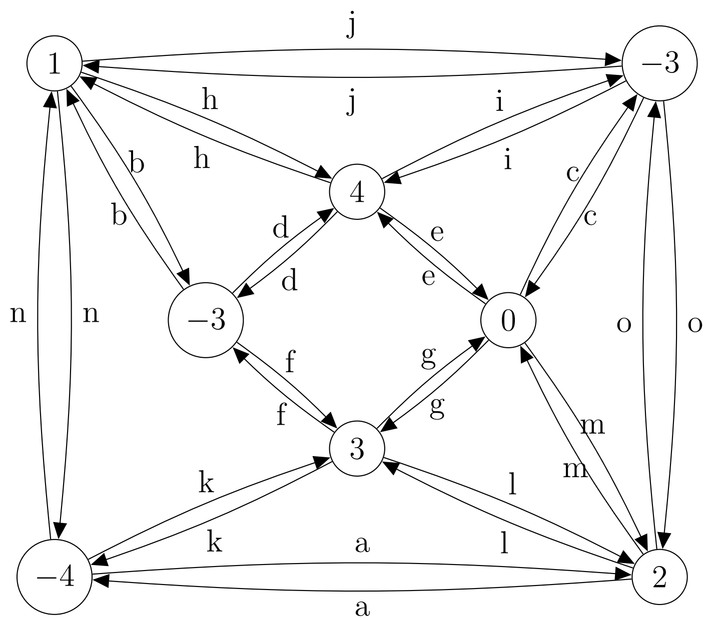
1: (*Try Min Cost Flow algorithm*)
Consider the following b-flow f in M.



Compute the cost of f.

Start computing the minimum cost b-flow by finding a sequence of augmenting cycles starting from f. (No need to use minimum mean cycles, do two augmentations. No need to solve it to optimality.)

You may use the following template to create residual graphs for finding the cycle.



2: (*Min Cost Flow as Linear Program*)
Solve minimum cost b-flow for M using linear programming. That is, formulate the problem using linear programming and solve it using Sage or APMonitor. Then draw the resulting b-flow.

3: (*Max Flow* $\subset$ *Min Cost Flow*)
Show that the Maximum Flow Problem can be regarded as a special case of the Minimum Cost Flow problem. That is, for an instance of Maximum Flow Problem find a reformulation to Minimum Cost Flow problem whose solution can be interpreted as a solution to Maximum Flow Problem. That is, find *simple* algorithm that is solving Maximum Flow Problem and using Minimum Cost Flow as a black box subroutine once.

4: (*Directed Minimum Mean Cycle*)
Implement Directed Minimum Mean Cycle in Sage. Before implementing the algorithm, show that it is possible to slightly modify the algorithm. Instead of adding an extra vertex s and edges from s to all other vertices, it is possible to simply assign $F_0(v) = 0$ for all $v \in V$ at the beginning. This avoids the hassle with adding an extra vertex. But it requires an argument that the algorithm is still correct.
Feel free to use any part of the Sage template. If you don't like the outline I made, don't use it.